Revista Iberoamericana de Automática e Informática Industrial

Revista Iberoamericana de Automática e Informática Industrial 20 (2023) 303-314

www.revista-riai.org

Una propuesta docente para el desarrollo de competencias en programación de robots manipuladores.

Fernando Gómez-Bravo*, Rafael López de Ahumada, Raúl Jiménez-Naharro.

Departamento de Ingeniería Electrónica, de Sistemas Informáticos y Automática, Escuela Técnica Superior de Ingeniería, Universidad de Huelva, Avda. de las Fuerzas Armadas, s/n. 21007, Huelva, España.

To cite this article: Gómez-Bravo, F., López de Ahumada, R., Jiménez-Naharro, R. 2023. A teaching approach for the development of skills in programming manipulator robots. Revista Iberoamericana de Automática e Informática Industrial 20, 303-314. https://doi.org/10.4995/riai.2023.18756

Resumen

Este artículo describe una propuesta docente para la formación de alumnos universitarios en técnicas de programación de robots manipuladores. El trabajo propone un conjunto de competencias básicas que agrupan las capacidades necesarias para el aprendizaje de este tipo de metodologías. El desarrollo de la estrategia docente se basa en la configuración de un sistema que permite, desde un mismo terminal, testar un programa bien en simulación o, de forma alternativa, en un robot real colaborativo. El método propuesto permite compartir entre el alumnado, de forma racionalizada, el acceso a los robots del laboratorio, permitiendo que el programa probado en simulación presente una estructura idéntica al implantado en el robot real. La propuesta se desarrolla en el entorno del programa MATLAB y el programa de simulación de robots RoboDK.

Palabras clave: Robots manipuladores, programación de robots, modelado y simulación, RoboDK, educación en robótica.

A teaching approach for the development of skills in programming manipulator robots

Abstract

This work describes a teaching proposal for training university students in programing robotic arms. The paper proposes a set of basic competencies that group the necessary skills for learning this type of methodology. The development of the teaching strategy is based on a system that allows testing a program from the same terminal, either in simulation or alternatively, in a collaborative real robot. The proposed method allows students to share, in a rationalized way, access to the laboratory robots, allowing the program to be tested in simulation to have an identical structure to that implemented in the real robot. The proposal is developed by using the MATLAB framework, and RoboDK, a program for robot simulation.

Keywords: Robotic arms, robot programming, modelling, and simulation, RoboDK, education in robotics.

1. Introducción

En la actualidad, cada día es mayor la presencia de asignaturas de robótica en los planes de estudios de las titulaciones técnicas, siendo necesario que ingenieros de distintas especialidades tengan formación en el campo de los sistemas robóticos y automáticos (Ajwad et al., 2017), (de la Peña et al., 2022). Particularmente interesante resulta la formación en el área de manipulación robótica, ya que son innumerables las actividades, de distintos ámbitos de la ingeniería, que se están automatizando bajo el paraguas del uso de manipuladores colaborativos (Bauzano et al., 2010). En este sentido, la formación en competencias para la programación de robots manipuladores cobra un gran interés.

Por una parte, esta formación contribuye al desarrollo general de destrezas vinculadas a la programación de sistemas automáticos (Valera et al., 2014), (de la Peña et al., 2022) y al conocimiento adquirido en el campo teórico de la robótica. Por otra, proporciona una capacitación de carácter práctico para desarrollar aplicaciones robóticas que pudieran aparecer en el transcurso del desempeño profesional. En consecuencia, resulta de interés analizar las competencias adquiridas cuando se incluye la programación de manipuladores como actividad práctica en los planes de formación de los estudios técnicos

* Autor para correspondencia: fernando.gomez@diesia.uhu.es

superiores, para posteriormente analizar las características que han de tener las estrategias docentes que las desarrollen.

En este sentido, existen diversos estudios que abordan el desarrollo de destrezas específicas en el proceso de aprendizaje de programación de robots (Chih-Chien Hu et al, 2020), (Mondad et al., 2009) aunque gran parte de ellos se circunscriben al entorno de las enseñanzas medias (Newton et al., 2013), (Huang et al, 2013), planteando el análisis de la programación en robótica desde un punto de vista general. Hay otros trabajos en los que se particularizan aspectos relativos a la programación de manipuladores en enseñanzas superiores, (Gil et al., 2015), (Ajwad et al., 2017). En ellos se presta atención a la descripción de herramientas desarrolladas para la simulación y programación de este tipo de robots.

En particular, el presente artículo atiende expresamente a esa última cuestión: plantea un conjunto de competencias asociadas al desarrollo de actividades prácticas relativas a la programación de manipuladores, y propone una estrategia docente para desarrollarlas. Para ello, se analizan las competencias específicas y los resultados de aprendizaje más frecuentemente encontrados en programas de asignaturas universitarias relacionadas con la robótica industrial, y se proponen un conjunto de competencias básicas que servirán de apoyo para el desarrollo de las destrezas necesarias para programar manipuladores robóticos.

Desde esta perspectiva, el desarrollo de competencias en programación textual de robots cobra especial relevancia. Esta estrategia de programación representa una solución muy versátil y, además, no solo aparece como una alternativa a las técnicas de programación guiada o las basadas en herramientas gráficas, sino que también representa un complemento eficaz para las mismas. Por este motivo, dicha estrategia cobra especial protagonismo en la propuesta que presenta este trabajo.

En cualquier caso, para que el proceso de aprendizaje sea totalmente eficiente, es muy recomendable que el alumnado, además de las clásicas lecciones teóricas de robótica, que implican, entre otras cuestiones, el estudio de la representación de la pose, la cinemática, la dinámica y los métodos de control del robot, aborde aspectos de implantación prácticos en los que se planteen la realización de tareas y operaciones para las que han sido pensados los robots manipuladores. Estos aspectos deben incluir tanto las problemáticas que aparecen en la especificación física de la tarea (tales como definición de agarres, posicionamiento de herramienta, definición del espacio de la tarea etc.) como los vinculados a la definición de movimientos, control de ejecución o coordinación con otros sistemas. En cualquier caso, el planteamiento docente debe tener una orientación práctica, en el sentido de ser planteado como problemas donde el alumnado se acostumbre a proyectar el conocimiento adquirido para encontrar soluciones a cuestiones concretas sobre manipulación robótica.

Esta perspectiva, que comparte la vocación práctica del proceso de aprendizaje con innumerables áreas docentes de la ingeniería, ha cobrado, en los últimos años, un particular protagonismo en el campo de la robótica, para el que se han desarrollado numerosas herramientas de simulación que permiten al alumnado la verificación, en un escenario dinámico, de las principales cuestiones abordadas en el bloque teórico de la asignatura.

Existen diversas herramientas de simulación y programación proporcionada por los fabricantes de robots industriales, entre los que cabe mencionar: RobotStudio de ABB (Holubek et al., 2014), Kuka.Sim de KUKA (Lukač, 2018) o Staübly Robotics Suit de Staübly (Alhama Blanco et al., 2018). Obviamente estas herramientas están optimizadas para el desarrollo profesional en el ámbito tecnológico del fabricante y no suelen cubrir aspectos académicos. Igualmente, se han desarrollado entornos de simulación abiertos al uso manipuladores de distintas tecnologías como VREP (Pires et al., 2019), RoboDK (Chakraborty et al., 2021) o GAZEBO (Vivas y Sabater, 2021). En cuanto al contexto académico merece la pena destacar las herramientas desarrolladas en el entorno de MATLAB tales como HEMERO (Maza y Ollero, 2001), la toolbox de Peter Corke (Corke, 2017) o ARTE (Gil et al., 2015). De estos tres últimos trabajos, los dos primeros cubren aspectos académicos vinculados con numerosas cuestiones de modelado y control de manipuladores, mientras que el tercero además incluye contenidos relacionados con la programación de manipuladores, particularmente emulando el lenguaje Rapid de ABB.

Es indudable que, en la actualidad, el uso de las herramientas de simulación resulta ventajoso, no solo en el proceso de aprendizaje, sino que también forma parte del proceder profesional habitual. No obstante, la experiencia indica que la comprobación en robots reales de la eficacia del código programado es una actividad significativa desde el punto de vista del aprendizaje. Por una parte, es necesario hacer partícipe al alumno de los aspectos de la realidad que son difíciles de recrear en simulación (particularidades del entorno, no idealidades de los componentes que conforman el robot, diferencia entre parámetros reales y simulados, etc.) y las consecuencias mayores o menores que esto puede tener a la hora de implantar en la realidad las estrategias simuladas. Por otra, la simulación soporta soluciones que en la realidad resultan poco prácticas o incluso inviables. Por tanto, la enseñanza de la robótica, una disciplina que desde sus inicios presenta una vocación aplicada muy acentuada, requiere de la interacción del alumnado con robot reales.

En este contexto, la propuesta docente que presenta este artículo presta especial atención a las actividades prácticas de programación textual de robots manipuladores, combinando de forma adecuada la utilización de simuladores y de robots reales, de forma que el proceder del alumno no se altere al cambiar la simulación por la interacción con el robot real. La propuesta no olvida la utilización de otras estrategias de programación, estando definida alrededor de las competencias inicialmente propuestas.

Junto a este planteamiento, la metodología desarrollada presta atención a diversos aspectos prácticos vinculados a su implantación. Como es muy habitual en los laboratorios docentes, el número de robots disponibles es menor que el de alumnos que asisten a un grupo de prácticas. Esto es muy natural cuando se trata de utilizar robots profesionales para los que concurren diversas razones limitantes en cuanto al número de robots disponibles: elevado precio, considerable espacio ocupado por el robot, costes de mantenimiento etc. Por tanto, es frecuente que sea necesario compartir el uso de estos equipos durante el periodo docente: bien se restringe el número

de alumnos que interaccionarán con los robots en cada sesión, o bien se planifican las sesiones de manera que los alumnos puedan alternar el uso del robot con la realización de otras actividades programadas. Es en esta última opción donde se ubica la propuesta presentada en el presente artículo.

En concreto, se propone un método que garantiza el acceso racionalizado a los robots reales de manera que sea posible la compartición ordenada de la infraestructura del laboratorio. Para ello, se ha diseñado un protocolo de manera que una parte del alumnado (previamente habilitada) pueda estar trabajando con los robots, mientras el resto trabaja con el simulador a la espera de ser habilitado su acceso a dichos robots.

Si bien esta estrategia ha sido ideada para la realización de prácticas presenciales, estando ubicado todo el alumnado en un mismo laboratorio y con presencia de personal docente, la arquitectura propuesta, podría ser utilizada para permitir el acceso remoto, aunque ello supondría realizar algunas adaptaciones específicas, en el desarrollo de la propuesta docente y en la configuración de acceso a la red.

El artículo está estructurado de la siguiente forma. Tras la introducción, en la sección 2 se proponen un grupo de competencias básicas que engloban diversas capacidades que deben ser adquiridas al realizar actividades de programación de robots manipuladores. En la sección 3 se dan detalles sobre la propuesta y las características técnicas del sistema diseñado para la implantación de la propuesta. En la sección 4 se presenta un caso de estudio real. Se describe la programación de las actividades prácticas desarrolladas dentro de una asignatura de robótica de grado, vinculándolas a las competencias descritas anteriormente. El artículo termina con las conclusiones y futuros trabajos.

2. Competencias básicas para la programación de robots manipuladores

Dentro del catálogo de titulaciones universitarias españolas, existen numerosos planes de estudio que incluyen asignaturas de introducción a la robótica o de robótica industrial. Para la definición de las competencias que se proponen en este artículo se han revisado numerosas guías docentes de este tipo de asignaturas, tanto en grados (Álvarez y Muñoz, 2022), (Gil y Sabater, 2022), (Gámez y Cano, 2022), (García y Serón, 2022), (González, 2022), (Lora y Castaño, 2022), (Romero y Gracias, 2022), como en másteres (Gamboa, 2022), (Cabanes y Mancisidor, 2022).

En el caso de los títulos pertenecientes a la familia industrial, la competencia específica más citada es: "Conocimientos de principios y aplicaciones de los sistemas robotizados"; que coincide con una de las competencias listadas en la Orden CIN/351/2009, (BOE de 20 de febrero de 2009) sobre requisitos para la verificación de los títulos de Grado que habiliten para el ejercicio de la profesión de Ingeniero Técnico Industrial. Por otro lado, en el ámbito de las titulaciones especialistas en robótica las competencias específicas son más diversas, ya que los programas suelen tener, o bien una orientación más computacional (Martín, 2022), u otra más vinculada al desarrollo mecatrónico (Galindo y Gonzalo, 2022). No obstante, como denominador común, en todas las asignaturas aparecen competencias específicas y contenido vinculados con la programación y la simulación de sistemas robóticos. Como dato significativo, en el conjunto de las guías analizadas, los resultados del aprendizaje incluyen objetivos tales como: ser capaz de utilizar sistemas para la simulación y la programación fuera de línea; aprender un lenguaje de programación robótico y ser capaz de utilizarlo en un brazo robótico real, u otros muy similares. El análisis de programas docentes de otras universidades no españolas arroja un resultado muy similar (Lu, 2019). A idénticas conclusiones se llega si se consultan libros de texto tradicionales tales como: (Ollero, 2001), (Torres et al, 2002), (Barrientos et al., 2007), (Craig, 2022). En todos ellos se dedican capítulos o secciones específicas al estudio de lenguajes y estrategias de programación de robots manipuladores.

En casi todos los programas de asignaturas estudiados se repiten dos hechos que justifican el trabajo que aquí se describe: a) la programación e interacción con robots reales aparece como eje vertebrador de las actividades prácticas planificadas; b) las competencias y los objetivos de aprendizaje específicos están descritos muy genéricamente, sin detallar destrezas básicas sobre las que se construirán los mismos. Por estos dos motivos, este trabajo pretende, por una parte, prestar especial atención al marco de actividades prácticas vinculadas al desarrollo de habilidades para programar robots manipuladores. Y por otra, establecer un conjunto de competencias (que en adelante se denominarán básicas, para diferenciarlas de las específicas), que maticen y particularicen aquellas destrezas sobre las que alcanzar las competencias específicas y los resultados del aprendizaje explicitados en los programas de las asignaturas.

Aunque esta propuesta centra su atención en la programación de robots manipuladores, los autores no pretenden olvidar la relevancia de otros contenidos. Todo lo contrario, este trabajo propone utilizar la transversalidad asociada a la programación de tareas de manipulación, para poder resaltar en las actividades prácticas la importancia del resto de cuestiones abordadas en las clases de teoría. Las competencias básicas propuestas se enumeran a continuación.

I. Capacidad para transformar relaciones lógicas funcionales en relaciones vinculadas a la geometría espacial: La definición y configuración de tareas de manipulación ha de codificarse, desde un punto de vista estático, en términos de relaciones geométricas que definen el agarre de un objeto o la ubicación de una herramienta sobre un elemento a mecanizar. Por otro lado, desde un punto de vista dinámico, la manipulación se traduce en trayectorias espaciales que han de cumplir diversos tipos de restricciones. Esta codificación es fundamental para el desarrollo de programas eficaces y, en consecuencia, el alumnado debe ser capaz de caracterizar matemáticamente estos elementos.

II. Capacidad para transferir conocimientos teóricos del campo de la robótica a una aplicación real: El 'corpus' teórico de la robótica de manipulación está bien definido en infinidad de textos docentes (Ollero, 2001), (Torres et al, 2002), (Barrientos et al., 2007), (Corke, 2017), (Craig, 2022). donde se describen, entre otros, detalles sobre cinemática, dinámica del robot, técnicas de planificación de movimientos, estrategias de control etc. Todos estos elementos se conjugan de forma activa cuando se plantea la ejecución de una tarea de

manipulación. Así, el alumnado deberá ser competente en la detección y solución de los problemas estudiados en teoría cuando se trate de robotizar una operación industrial en concreto. Esta proyección en la realidad de los conceptos abordados en las sesiones teóricas redundará en la propedéutica de la formación recibida, para, en un futuro, ejercer como profesional en este u otros ámbitos de la tecnología.

III. Capacidad para caracterizar formalmente la tarea de manipulación: En un principio, las operaciones de manipulación pueden ser entendidas de forma intuitiva, apelando a la experiencia diaria que de ellas tienen los seres humanos. Sin embargo, la programación de éstas en un robot requiere conceptualizar las relaciones entre el robot y el objeto a manipular o mecanizar, establecer la secuencia de ejecución, los condicionantes que derivan en la realización de una u otra acción, y estudiar los efectos que sobre la ejecución de la tarea tiene el entorno de trabajo. Esta capacidad engloba el conocimiento específico sobre operaciones típicas de la robótica industrial tales como: 'pick & place', ensamblado, trazado, inspección etc. y suelen tener como resultado la destreza para diseñar diagramas de flujo y definición de configuraciones espaciales que servirán de base para confeccionar el programa final.

IV. Capacidad para implementar la secuencia diseño -> simulación -> realidad: En la actualidad, el desarrollo de aplicaciones en distintos campos profesionales requiere de un proceso de verificación previo a la implementación real. Esta metodología ha venido utilizándose con frecuencia en el contexto de los sistemas de control y la automática, basándose en el desarrollo de simuladores. Precisamente la aparición del concepto de gemelos digitales ha causado que el mundo de la automatización, y la robótica en particular, evolucione hacia la utilización de sistemas software más complejos, donde sea posible testar los programas de una forma realista manteniendo la misma estructura que cuando sean implantados en los robots reales. En consecuencia, el alumnado debe ser capaz de diseñar programas que puedan testarse y depurarse virtualmente en una plataforma de simulación, para luego verificar su ejecución eficaz en un robot real.

Cualquier método que se plantee para enseñar a programar robots manipuladores deberá propiciar el desarrollo de estas competencias. Sin duda existen experiencias docentes que, en base a las herramientas de programación y simulación antes referenciadas, permiten el desarrollo de dichas capacidades. Sin embargo, suele haber una discontinuidad, en la forma de trabajo, entre las actividades que promueven las competencias I y II, más enfocadas en la codificación y el análisis matemático de los problemas de manipulación, y las destinadas a desarrollar las competencias III y IV, ligadas a la plataforma robótica y al lenguaje de programación utilizados. Por ejemplo, suele ser frecuentemente difícil verificar de forma directa, en la plataforma real, los resultados cuantitativos obtenidos en el desarrollo teórico. En este sentido, la propuesta que se describe en este artículo permite el planteamiento de actividades que fomentan la adquisición de estas cuatro competencias, garantizando una continuidad operativa en el desarrollo de todas ellas. Adicionalmente, los entornos de simulación y programación mencionados no suelen estar especialmente diseñados para que el acceso al robot se haga de una forma racional, beneficiando la compartición del uso del mismo. En relación con este aspecto, el método propuesto establece un procedimiento para que el alumnado tenga acceso al robot, de una forma coordinada por el profesor, sin que sea necesario cambiar de plataforma de programación si se están trabajando aspectos teóricos, de simulación o de implantación en el robot real.

3. Descripción de la propuesta docente

La metodología que se propone a continuación, al basarse en el uso del programa de computación científica MATLAB, puede desarrollarse también utilizando las herramientas descritas en (Corke, 2017) o en (Gil et al.,2015). No obstante, se ha optado por la implementación que se describe a continuación debido a la representación de robots tan realista que ofrece RoboDK y la larga lista de manipuladores soportados por este programa.

En particular, la propuesta permite programar la ejecución de tareas de manipulación robótica mediante un lenguaje de propósito general desarrollado en MATLAB. Mediante este lenguaje es posible simular la tarea programada o ejecutarla en el robot real. El desarrollo de este lenguaje permite la consecución de varios objetivos relacionados con las competencias detalladas. Concretamente combina sentencias vinculadas con los problemas estudiados en la parte teórica de la asignatura (competencias I y II) junto con comandos típicos de control de flujo y movimiento (competencias III y IV), haciendo posible así una continuidad operativa en actividades que desarrollen cualquiera de las cuatro competencias. De este modo, es posible vincular directamente las actividades de programación con los contenidos teóricos estudiados, facilitando la compresión de los procesos de configuración del robot y permitiendo manejar datos vinculados al estado del manipulador de una manera eficiente. Adicionalmente, el uso de este lenguaje pone de manifiesto al alumnado que, salvando ciertas peculiaridades características de cada fabricante, los comandos de control de flujo y de control de movimiento presentan gran similitud.

En el ejemplo que se muestra en este artículo se han utilizado dos manipuladores UR3 de la empresa Universal Robots (UR), que se encuentran en los laboratorios del departamento de Ingeniería Electrónica de Sistemas Informáticos y Automática (DIESIA) de la Universidad de Huelva, (ver Figura 1).

Figura 1: Laboratorio Docente de Robótica (DIESIA, UHU)

Los comandos del lenguaje desarrollado emulan, en cuanto a los argumentos que reciben, a los de control de movimientos y entrada/salida, propios de los robots de UR (URscript). Respecto a las sentencias relacionadas con los problemas teóricos, se han añadido comandos vinculados con el modelo directo e inverso del manipulador, con operaciones relativas a las matrices de transformación y con la conversión entre distintos formatos para representar la pose (ángulos de Euler, representación eje/ángulo, cuaternios etc).

A pesar de que la experiencia aquí presentada está muy orientada al uso de manipuladores de una marca muy concreta, la estrategia diseñada puede ser aplicada a cualquier tipo de manipuladores que presenten las mismas características de conectividad que los robots de UR. Además, la sintaxis del lenguaje permite aproximar la mayoría de los comandos de control de movimiento y flujo de cualquier otro lenguaje de programación de manipuladores como pueden ser VAL 3, RAPID o ACL.

En cuanto a la capacidad para simular la tarea programada, la característica más relevante del lenguaje desarrollado es que puede interaccionar con manipuladores simulados en el programa RoboDK. Este hecho dota al sistema propuesto de mayor versatilidad que si para simular se hubiese utilizado MATLAB, ya que RoboDK es una plataforma con una interfaz gráfica de gran realismo, que permite incorporar fácilmente al entorno de simulación, elementos periféricos como cintas transportadoras, herramientas, objetos para manipular u otros elementos diseñados externamente con un programa de CAD.

Así, la arquitectura del sistema desarrollado está pensada para hacer posible que, por una parte, el alumnado tenga acceso al uso de robots reales, compartiendo de forma segura el uso de los mismos, y por otra aprenda a desarrollar estrategias de manipulación, depurando el código programado mediante el uso de una simulación realista.

La forma en que se ha materializado este planteamiento de trabajo dual, la operatividad y fundamento del mismo y algún ejemplo de aplicación, se muestran en las siguientes secciones.

3.1 Características y requisitos

El sistema ha sido diseñado para hacer posible que el alumnado asistente a las sesiones prácticas pueda acceder de forma ordenada a los dos manipuladores UR3. El laboratorio dispone de un total de 10 puestos de trabajo, dotado cada uno de un computador (Figura 1).

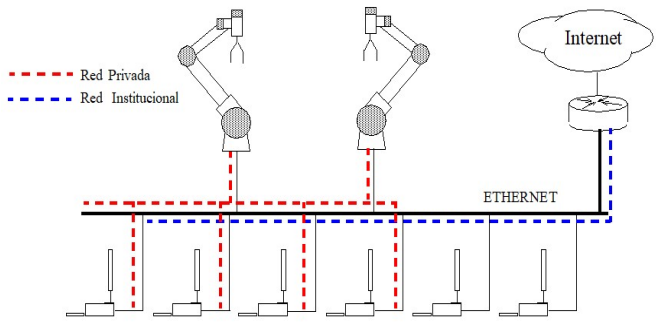


Figura 2: Esquema de conectividad

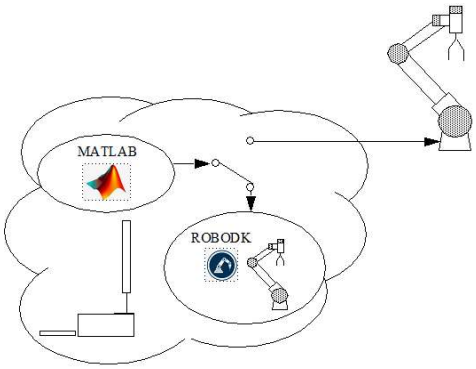


Figura 3: Configuración de la estación de trabajo

Desde cualquier computador es posible telecontrolar cualquiera de los dos robots o descargar un programa en ellos (ver Figura 2).

Es obvio que para racionalizar el uso de la instalación es necesario tener en cuenta los siguientes principios básicos:

- Cuando un grupo de estudiantes tenga acceso a un robot, este acceso será exclusivo, es decir, no será posible que ningún otro usuario acceda al mismo, impidiéndose que dos grupos interaccionen con el mismo robot al mismo tiempo.
- Es necesario que mientras los robots estén ocupados el resto de los grupos pueda depurar el código o realizar alguna otra actividad vinculada con la programación textual de robots, por lo que es conveniente realizar en paralelo trabajos con el robot real y el robot simulado.
- La ubicación física del alumnado, o su forma de proceder en el laboratorio deberá ser la misma tanto si trabaja con el simulador como si lo hace con el robot real.
- El programa que se esté testando debe ser esencialmente el mismo tanto si se trabaja con el robot real como si se trabaja con el robot simulado.
- La conectividad entre terminales y robots no debe alterar la conectividad habitual con el exterior de la red del laboratorio.

A partir de estos requisitos se configura cada puesto de trabajo de la forma que se observa en la Figura 3. Cada estación de trabajo está compuesta por un computador personal donde se ejecuta Windows 11, y en el que se ha instalado la versión 2020 de MATLAB y la versión 5.0 de RoboDK. El computador está conectado a la red interna de la universidad mediante una interfaz física que está configurada con dos interfaces lógicas. Una de ellas permite acceder a la red institucional y la otra a una red privada que se encuentra configurada dentro de la misma red física (Figura 2).

La configuración propuesta permite que el alumnado, desde el mismo terminal, mantenga conectividad con el exterior para acceder a los servicios de Internet y, al mismo tiempo, con los robots reales configurados como terminales de la red privada. Como se mostrará en los próximos apartados, la comunicación con el robot se ha implementado utilizando MATLAB, desde donde el alumnado podrá interactuar de forma alternativa con el robot real (a través de la infraestructura de red) o con el robot simulado en RoboDK.

Las interfaces de comunicación entre MATLAB, el robot y RoboDK se describen en los siguientes apartados.

3.2 El lenguaje desarrollado

Dado que MATLAB es un lenguaje interpretado, la estructura esencial de los programas que se desarrollan, es muy similar a la estructura de los programas utilizados para programar los robots de Universal Robots. La definición y los tipos de variables en ambos entornos son semejantes, y, como era de esperar, las sentencias de control de flujo también guardan gran similitud. Además, la versatilidad de MATLAB hace que el desarrollo del software sea lo suficientemente flexible como para emular también los comandos de otros lenguajes.

En relación con los sistemas de comunicación, MATLAB posee un amplio espectro de comandos para definir puertos vinculados a los protocolos de comunicación más utilizados en el mundo de la industria. Adicionalmente, la definición de interrupciones ligadas a eventos puede asimilarse al tratamiento que hace MATLAB de eventos asíncronos mediante funciones 'callback'. De esta forma, las competencias de programación que se desarrollan en este entorno son de naturaleza idéntica a las necesarias para diseñar programas de control en las distintas plataformas de programación de manipuladores robóticos existentes en el mercado.

En particular, el lenguaje diseñado se basa en el desarrollo específico de comandos y funciones vinculados al control de movimiento y a la gestión de entradas, salidas y comunicaciones.

Desde el punto de vista de control del movimiento, se ha desarrollado un comando correspondiente al movimiento punto a punto mediante la especificación articular, o de la pose (posición y ángulos de Euler) de la herramienta instalada en el manipulador (pinza de OnRobot). Igualmente se han generado comandos para el movimiento en línea recta hacia un punto determinado o hacia una dirección. También se han desarrollado comandos específicos para el accionamiento de la pinza y para el manejo de las entradas/salidas digitales y analógicas. De igual modo se han definido comandos para la gestión de los registros del servidor Modbus del que dispone el manipulador. Desde el punto de vista de la cinemática directa, se ha implementado una función que devuelve la matriz de transformación asociada a la herramienta para una configuración articular dada. Esta función permite escoger la opción de generar una ventana gráfica de MATLAB con la representación del manipulador en la configuración articular especificada. En cuanto a la cinemática inversa, se ha desarrollado una función que proporciona la configuración articular para una pose de la herramienta determinada. Particularmente, esta función, además de la especificación de la pose, contempla la selección de cada una de las ocho posibles configuraciones inversas: codo arriba, codo abajo, muñeca adelante, muñeca atrás, para configuración de brazo derecho y lo mismo para la configuración de brazo izquierdo

3.3 La interfaz con el robot

El controlador de los robots UR proporciona un conjunto de servidores que permiten acceder al estado del robot y aceptar la recepción de comandos pertenecientes al lenguaje URscript (ver Figura 4).

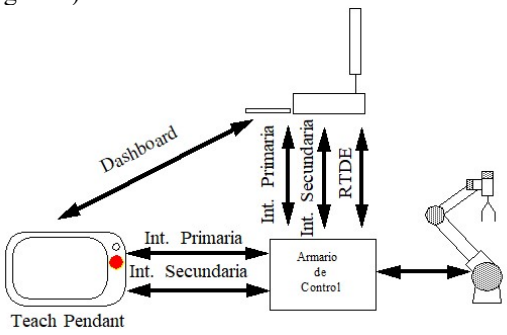


Figura 4: Arquitectura de servidores UR.

Todos los servidores están soportados sobre TCP/IP utilizando Ethernet, y tienen asignado un puerto de escucha particular. Básicamente existen tres servidores, conocidos como interfaz primaria, secundaria e interfaz en tiempo real (RTDE). La diferencia entre la interfaz primaria y secundaria reside en que desde la primaria además del estado del robot también se transmiten un conjunto de mensajes complementarios. La frecuencia de actualización de datos en estas interfaces es de 10Hz. El RTDE tiene un comportamiento similar a la interfaz secundaria pero la velocidad de actualización de la información se realiza a una frecuencia de 500Hz en el caso de la serie 'e' y 125Hz en el caso de la serie 'CB'.

Las interfaces primaria y secundaria son utilizadas por el 'Teach Pendant' de UR (una 'tablet' con pantalla táctil) para recibir información del robot y transmitirle al controlador los programas realizados. La interfaz para control en tiempo real permite el control remoto del manipulador desde una CPU externa, bien un PLC o un computador.

En el trabajo presentado en este artículo se utiliza el servidor RTDE, cuyo puerto de escucha TCP es el 30003. Este servidor acepta tanto comandos individuales, codificados en código ASCII, para ser ejecutados por el controlador, como programas completos. Las funciones implementadas transforman la información recibida en comandos del lenguaje URscript y los envía al servidor del robot correspondiente, recibiendo un mensaje de aceptación que confirma la recepción de la información. Igualmente existe la posibilidad de agrupar todos los mensajes contenidos en el programa de MATLAB y transformarlos en un programa de UR.

De esta forma el sistema desarrollado hace posible trabajar de dos formas distintas. Por una parte, permite que el programa desarrollado sea ejecutado desde MATLAB, realizando la llamada secuencial de los comandos correspondientes y consultando el estado del robot a lo largo de la ejecución para controlar el flujo del código generado. Y por otra, es posible generar un programa completo y enviarlo al controlador para que se ejecute localmente.

3.4 La interfaz con RoboDK

RoboDK es un programa de simulación de sistemas robóticos industriales, que proporciona una interfaz gráfica 3D de apariencia muy realista (ver Figura 5). Ya que los modelos cinemáticos y gráficos que incorpora se aproximan con gran

precisión a la realidad. RoboDK permite programar tareas robóticas de diversa índole proponiendo una metodología particular.

Además, si existe conexión con el robot, la aplicación permite interaccionar remotamente con éste, haciendo posible que el programa diseñado se ejecute en el mismo robot. No obstante, la metodología de programación propuesta por RoboDK presenta algunos inconvenientes si se compara con la flexibilidad que aporta la programación textual. Por una parte, la interfaz de RoboDK no permite la definición de bucles iterativos para las secuencias de movimiento, lo que implica que la repetición de movimientos similares, donde cambien los puntos de paso, exija la definición explicita de estos puntos, no permitiendo que los mismos se redefinan, por ejemplo, dentro de una estructura de bucle 'FOR'.

Además, existe otra desventaja frente a la programación textual. En concreto, para modificar las configuraciones espaciales, tanto la de los puntos de paso como la de los objetos a manipular, es necesario acceder a un menú particular. Mientras que, en la programación textual, estas modificaciones se hacen directamente en las líneas de código reservadas para la definición de configuraciones. Por estos motivos, en el presente trabajo, RoboDK solo se utiliza para testar y depurar el código desarrollado en MATLAB antes de ser implantado en el robot real.

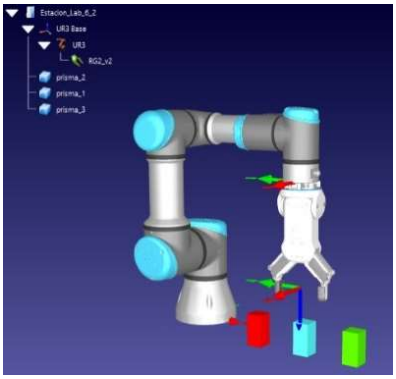


Figura 5: Escenario de simulación RoboDK.

RoboDK ofrece una interfaz de programación de aplicaciones (API) con MATLAB, de manera que es posible ejecutar código en MATLAB que implique la activación, desactivación y manejo de los elementos definidos en una estación dentro de un fichero RoboDK. Mediante esta API cualquier elemento en el árbol de la estación de RoboDK puede ser recuperado, mediante un objeto, en el entorno de ejecución de MATLAB, siendo posible realizar diferentes operaciones sobre los elementos vinculados a dicho objeto.

Gracias a esta forma de trabajo, los comandos de movimiento definidos en el lenguaje de programación pueden actuar sobre los objetos de una estación RoboDK. Dichas actuaciones suelen ser: la ejecución de un movimiento del robot, la activación de la pinza, o recabar información sobre algún objeto o sobre el estado del robot.

Una vez el código esté depurado, será momento de ser probado en el robot real. El mecanismo por el cual un comando interacciona con una plataforma u otra se describe en la siguiente sección.

3.5 Integración del sistema

Para conseguir que MATLAB pueda interaccionar con el robot o con el programa de simulación se ha programado una función llamada 'Iniciación()', que proporciona la conexión correspondiente. Su prototipo es:

```
Identificador = Iniciacion(maquina,codigo)
```

Esta función tiene como misión establecer la conexión con la plataforma especificada, inicializando las variables correspondientes que, o bien utilizan la API de RoboDK, o definen el puerto TCP con el que se realizará la conexión con el robot accediendo al servidor RTDE.

La variable maquina es una cadena de caracteres que podrá tomar los valores: 'RBDK'; 'Robot_1'; 'Robot_2', según se vaya a realizar una conexión con RoboDK o con uno u otro robot. La variable código se corresponde con un código decimal de dos dígitos.

El código es asignado por el profesor para cada pareja, de forma que la identifica al tratar de conectar con el robot.

La Figura 6 representa la secuencia de establecimiento de conexión de la estación de trabajo tanto con el robot, como con RoboDK.

Para el caso de comunicarse con el robot, la función espera recibir una confirmación de conexión con éste si el valor de la variable código coincide con el código configurado en el robot por el profesor. Si el código enviado es correcto, el proceso de iniciación se hará efectivo, y la función devolverá un identificador que está relacionado con la plataforma con la que se desea trabajar. De lo contrario, el identificador estará relacionado con un mensaje de error asociado a la falta de conexión.

Para el caso de comunicación con la API de RoboDK, el código enviado es irrelevante, y se devuelve el identificador apropiado de conexión si se ha establecido correctamente la conexión con el programa. En caso contrario se devuelve un código de error.

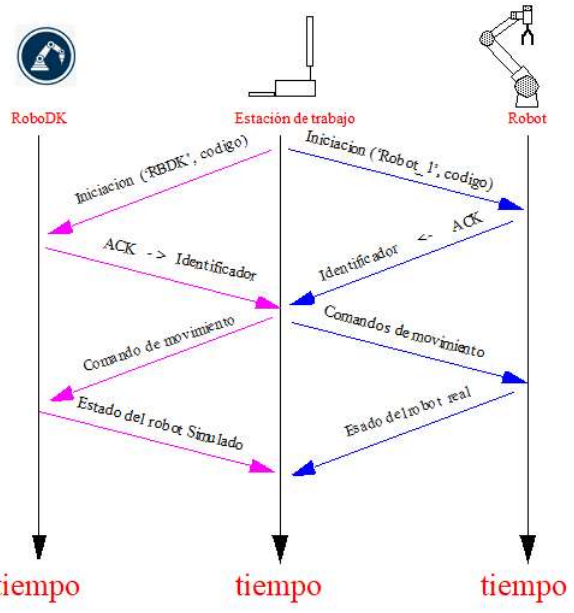


Figura 6: Conexión de un terminal de trabajo con RoboDK y el robot.

Previamente a que un grupo se conecte con un robot, el profesor ejecuta en su terminal la función 'mandar_codigo ()', cuyo prototipo es:

```
output=mandar_codigo(maquina,codigo)
```

En esta función, la variable máquina es una cadena de caracteres que podrá tomar los valores: 'Robot_1'; 'Robot_2' y la variable código se corresponde con el código asociado al grupo que en ese momento esté autorizado a trabajar con el robot.

La Figura 7 representa la secuencia de establecimiento de conexión de la estación de trabajo del profesor con el robot para configurar el código, así como el intento de conexión con el robot desde dos estaciones de trabajo diferentes, donde una de las peticiones de conexión es denegada.

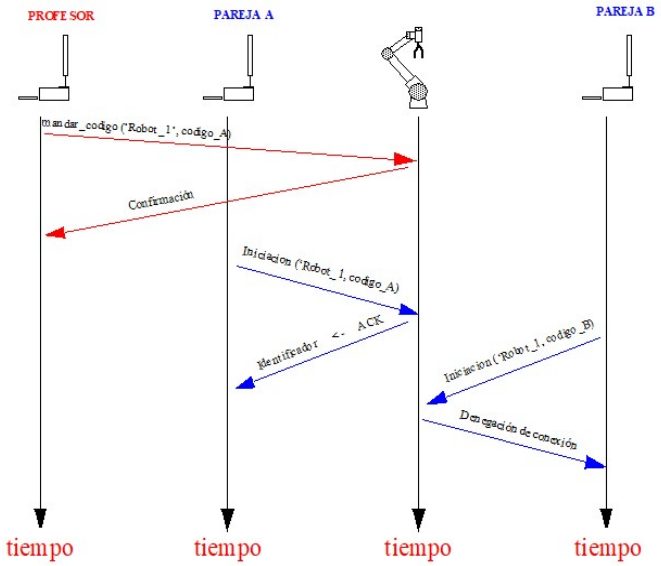


Figura 7: Conexión entre el terminal del profesor y el robot

Una vez invocada la función 'mandar_codigo()' desde la estación del profesor, ésta devuelve un valor que indica si la conexión con el robot se ha realizado con éxito o no. Si la conexión con el robot especificado ha sido correcta, el robot almacenará un valor correspondiente al código configurado. En caso de que la conexión con el robot no haya sido posible la función devuelve el código de error correspondiente, lo que significará que el robot no ha almacenado el código enviado.

Cuando un grupo de alumnos quiera interaccionar con un robot, y trate de conectarse con el mismo, utilizando el código que le ha sido asignado, la función 'Iniciación()' leerá el código almacenado en el robot. Si ambos códigos coinciden, la comunicación con el robot queda correctamente inicializada, se generará el identificador, y será posible seguir ejecutando el código programado. Si no coincide, el terminal con el que esté trabajando el grupo de alumnos recibirá un mensaje de error, indicando que no está autorizado a conectarse con el robot. Todos los comandos de ejecución de movimientos requieren entre sus argumentos el identificador y el código asignado, de forma que si en el transcurso de la ejecución del código el profesor estima oportuno retirar la autorización al grupo, bastará con que éste cambie el código activo en el robot para que los comandos programados dejen de ser efectivos.

Hay que advertir que este procedimiento no pretende servir como mecanismo de seguridad para evitar injerencias mal intencionadas en el uso del robot o ataques cibernéticos, sino que se utiliza como método para administrar racionalmente el uso del robot, evitando que errores de codificación o despistes del alumnado generen interferencia entre grupos de trabajo durante las pruebas con el robot real. El desarrollo de otros mecanismos de acceso que garanticen la seguridad integral del control de robot está fuera del objetivo del presente trabajo.

4. Caso de estudio

En este apartado se describe el desarrollo de un grupo de actividades vinculadas a las prácticas sobre manipulación robótica utilizadas en el transcurso de la asignatura Robótica y Automatización Industrial II. Se trata de una asignatura del Grado de Ingeniería en Electrónica Industrial de la Universidad de Huelva con una carga docente de 6 créditos ECTS (3. 2 teóricos y 1.8 prácticos). También se incluyen unas actividades con cierta similitud en la asignatura Robótica del Grado en Ingeniería Informática, de la misma universidad.

El curso dispone de 12 sesiones prácticas semanales: las dos primeras sesiones están dedicadas a actividades sobre buses industriales de comunicación, mientras que en las 10 sesiones restantes se abordan temas relacionados con la manipulación robótica. En estas actividades se trabajan conceptos vinculados las tareas de 'pick & place', el uso básico de RoboDK, la programación del robot utilizando el lenguaje desarrollado dentro del entorno MATLAB, y la programación utilizando el panel de control del robot y su interfaz gráfica.

Cada grupo de laboratorio está compuesto por 16 alumnos agrupados en parejas. Durante la semana 3 a la 5 se realizan actividades vinculadas con las competencias I y II. Las siete semanas siguientes están dedicadas a desarrollar las competencias III y IV. Para cada sesión se entrega con antelación, a través de la plataforma Moodle, una guía con información y documentación a cerca de las actividades a desarrollar y de los objetivos a alcanzar. El alumnado debe asistir a la sesión con la guía estudiada. A continuación, se describen estas diez sesiones y se dan detalles del procedimiento utilizado para su desarrollo.

Semana tercera: Definición y configuración del agarre. En esta sesión se trabajan conceptos relativos a la configuración del agarre en cuanto a posición relativa de la pinza respecto al objeto a manipular. Se estudian distintas estrategias de agarre y se propone un método para obtener la configuración de la herramienta a partir de los ángulos de Euler de la pieza a manipular y de la matriz de agarre definida. Así mismo, se plantea el trabajo con distintos tipos de representación de la pose. En esta sesión el alumnado utiliza el entorno gráfico de MATLAB para representar la posición relativa de una pinza virtual respecto a los bloques a manipular, ambos con sus sistemas de referencias locales (ver Figura 8).

Semanas cuarta y quinta: Modelos directo e inverso y trayectorias de transición. Para estas sesiones, en primer lugar, se propone obtener una representación gráfica del manipulador a partir de los parámetros Denavit-Hartenberg (estudiados previamente en las clases de teoría).

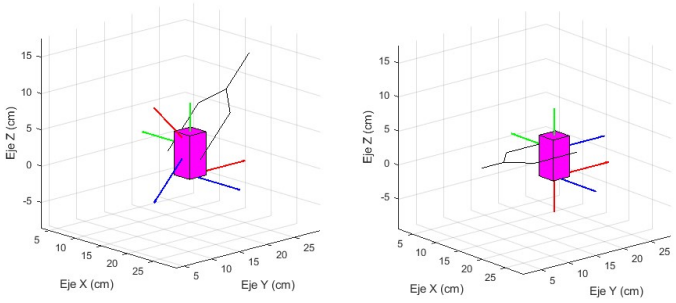
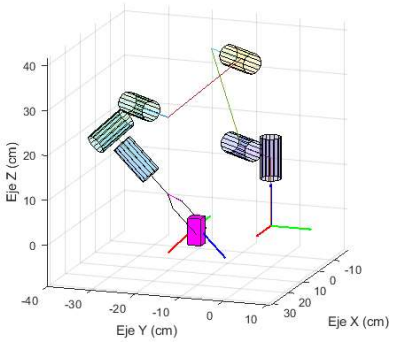
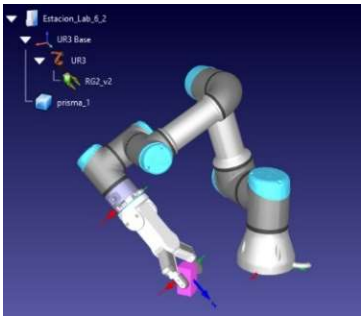


Figura 8: Estudio de la posición de agarre en distintas configuraciones

Esta representación se puede obtener utilizando la propia 'Robotics System Toolbox' de MATLAB, aunque en el caso mostrado en este artículo se utiliza una herramienta de representación desarrollada por el mismo equipo docente (ver Figura 9 a).



a)



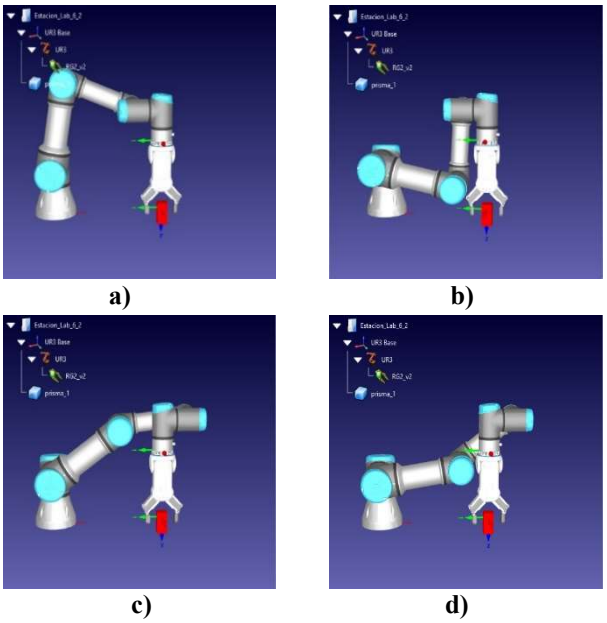
b)

Figura 9: Representación del robot: a) MATLAB; b) RoboDK

A partir de esta representación se propone utilizar los conocimientos adquiridos en clase sobre el modelo inverso para completar una función que lo resuelve. Con esta función, los datos de agarre de la sesión anterior y el modelo directo desarrollado, es posible representar el manipulador agarrando una pieza en la configuración de agarre deseada (ver Figura 9 a).

En la tercera sesión se introduce el programa RoboDK y su operatividad básica. De esta forma, transfiriendo los valores numéricos obtenidos con MATLAB al interfaz de RoboDK, es posible conseguir la misma representación, pero con la visualización realista de RoboDK (ver Figura 9 b). Esta sesión se aprovecha para abordar las posibles alternativas de resolución que tiene el problema cinemático inverso para una misma configuración de agarre; estudiando algunas soluciones utilizadas menos frecuentemente, como las de muñeca hacia detrás, codo hacia abajo etc. (ver Figura 10). También se utiliza la interfaz de usuario que ofrece RoboDK para establecer manualmente trayectorias de transición entre configuraciones espaciales arbitrarias y se enseña al alumnado a detectar distintos tipos de singularidades al realizar las trayectorias de transición. Durante la realización de esta actividad, se plantean cuestiones relacionadas con la cinemática y la dinámica del manipulador, que se resuelven cuantitativamente utilizando recursos para ser utilizados en MATLAB tales como HEMERO o ARTE.

Semanas sexta a octava: Programación, simulación e implantación en el robot I. Para estas sesiones, inicialmente se proporciona al alumnado un escenario RoboDK con un conjunto de bloques que se han de manipular con el fin de ensamblar una construcción predefinida (ver Figura 11 a). En la guía se introducen las bases del lenguaje de programación desarrollado y las estrategias básicas de programación.



a) b) c) d)

Figura 10: Distintas soluciones al problema cinemático inverso: a) codo arriba y muñeca hacia adelante; b) codo abajo y muñeca hacia adelante; c) codo arriba y muñeca hacia atrás; d) codo abajo y muñeca hacia atrás.

Los alumnos han de aplicar los conocimientos adquiridos en las sesiones anteriores y los correspondientes a las clases teóricas para generar un código que en primer lugar testarán en RoboDK (ver Figuras 11 b, c y d). Posteriormente, una vez el alumnado comprueba que el código permite realizar el ensamblado correctamente, se procede a interaccionar con el robot real (ver Figuras 11 e, f, g y h).

Para llegar al proceso de prueba final se realizan varias tentativas partiendo de versiones del problema más sencillas (con menos bloques y con una construcción más simple), hasta llegar a la versión final solicitada. De esta forma, el uso de los robots va compartiéndose a medida que el alumnado va completando la ejecución de la tarea en sucesivas versiones. Lo habitual es que el alumnado empiece a interaccionar con el robot hacia la séptima semana.

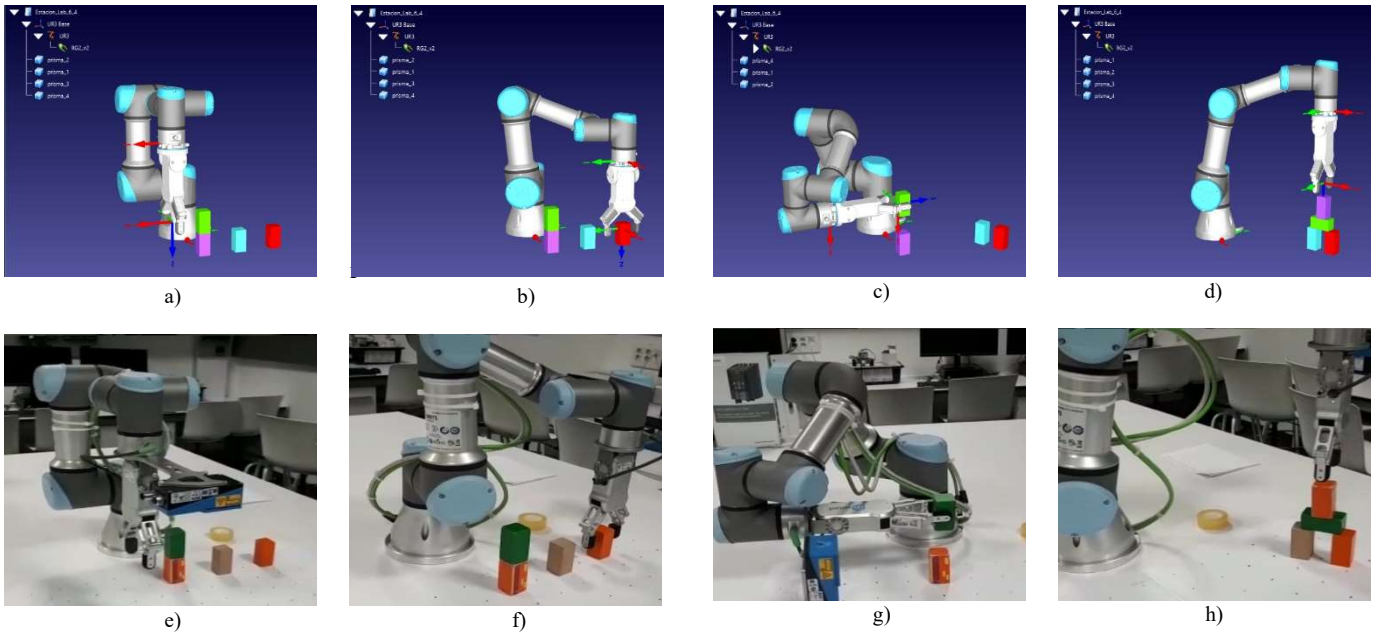


Figura 11: a) - d) operación de montaje simulada mediante RoboDK; e) - f) Verificación del programa en el robot real.

El tiempo que suele tardar un grupo en hacer que el código funcione correctamente en el robot suele ser de unos 5 minutos, ya que la depuración utilizando RoboDK ahorra mucho tiempo en la implementación real. Con el fin de racionalizar el uso compartido de los robots, al alumnado que teniendo testado su código no pueda acceder al robot, por estar éste ocupado, se le proponen modificaciones al programa elaborado mientras esperan el turno en el que el robot quede libre.

Semanas novena a duodécima: Programación, simulación e implantación en el robot II. Durante estas sesiones se propone al alumnado realizar un programa donde el robot ejecuta una operación similar a la anterior, pero esta vez el programa lo ha de codificar utilizando el propio panel del robot mediante la interfaz gráfica que éste ofrece. Para este caso, en el programa se incluye la lectura, por parte del robot, de una señal de control procedente de un PLC que se encuentra conectado mediante 'Profinet'. Cada pareja de alumnos tiene una sesión para realizar esta actividad. Al resto del alumnado, que no esté interactuando con el robot, se le propone la realización de un programa, utilizando el lenguaje desarrollado, donde el robot ha de realizar un trazado sobre una superficie, o una tarea de empaquetado. Los alumnos trabajan, simulando en RoboDK, y comprueban en el robot el funcionamiento de su programa en los momentos en que alguno de los robots quede libre.

Igualmente se propone a los alumnos utilizar herramientas como HEMERO, las toolboxes de Corke o ARTE, para simular estrategias de control que emulen el comportamiento del robot realizando la tarea programada.

Obviamente existen otras cuestiones de naturaleza práctica, como la instalación, mantenimiento y calibración del manipulador, que son contenidos relevantes en el currículo del alumnado. Su desarrollo, y el de las competencias en programación de robots es complementario y no excluyente. No obstante, la limitación del tiempo del que se dispone para las actividades de laboratorio condiciona su inclusión como actividades prácticas regulares, abordándose habitualmente este contenido entre las clases teóricas y prácticas, donde se introducen las particularidades técnicas vinculadas a estas cuestiones.

5. Análisis del impacto de la propuesta

El impacto de la metodología propuestas se ha evaluado mediante la realización de encuestas entre el alumnado, además de comparando las actividades propuestas en este trabajo con las actividades realizadas en cursos anteriores.

Dicha comparación atiende al grado de dificultad y al número de actividades de manipulación que realiza el alumnado al final del cuatrimestre.

En este sentido, hay que señalar que, según se desarrollaban anteriormente las actividades prácticas, no existía una integración directa entre el software de simulación y el robot, y el alumnado ocupaba completamente los robots mientras se depuraba el código de los programas, hasta conseguir el funcionamiento de la práctica. Con ese procedimiento el uso del robot no se compartía de forma tan efectiva, y como consecuencia, la mayoría del estudiantado solo completaba una práctica de "pick & place", de menor dificultad que la actividad de montaje mostrada en este artículo, y otra de programación utilizando el terminal del robot. En la actualidad, todo el alumnado realiza, como mínimo, la práctica de montaje y una práctica de trazado, además de la programación utilizando la interfaz propia del robot.

Adicionalmente, los grupos que avanzan con más rapidez realizan modificaciones de las actividades sugeridas, con lo que muchos acaban programando una tarea de empaquetado, o incluso un ensamblado de torres siguiendo las reglas del juego de torres de 'Hanoi', o incluso proponiendo un escenario de simulación propio que luego se replica en el robot real. Por tato, al aumentar el número de actividades de manipulación realizadas, así como su complejidad, puede afirmarse que la metodología propuesta beneficia el desarrollo de las competencias enunciadas en este artículo.

Respecto a las encuestas realizadas, en ellas se platean un total de 19 enunciados, con 5 posibles respuestas en las que el alumno expone su opinión respecto al enunciado. En concreto, las respuestas se corresponden con la valoración de 1 a 5, siendo 1 equivalente a la opinión 'muy en desacuerdo', y 5 a la opinión 'muy de acuerdo'. Se trata de una escala Likert (Heiberger y Robbins, 2014). Particularmente, los datos que se presentan a continuación se corresponden con las encuestas realizadas a dos grupos distintos (los índices de participación en cada grupo han sido del 70% y del 81%) siendo 22 el total de participantes (sumando el alumnado de ambos grupos).

La Figura 12 ilustra el análisis estadístico realizado sobre los resultados de las encuestas. En ella se muestran los diagramas de cajas correspondientes a cada uno de los enunciados. Se representan con segmentos rojos los valores de las medianas, y con cruces los valores atípicos. En trazo negro se representa la evolución del valor medio de las respuestas.

Pueden interpretarse los resultados de la encuesta atendiendo a cuatro áreas: a) Mejora de habilidades y conocimiento previos; b) percepción sobre el uso del simulador; c) percepción sobre el uso del robot real; d) percepción sobre el método seguido.

a) Mejora de habilidades y conocimiento previos. En relación con esta área merece la pena comentar los resultados del enunciado P1: "Tenía conocimientos previos de MATLAB". Analizando la gráfica correspondiente se aprecia cómo en el conjunto del alumnado, el conocimiento previo sobre MATLAB era variado. Aunque la mediana está en 4, y la media toma un valor alrededor de 3.5, se observa que hay un conjunto de estudiantes que están 'poco de acuerdo' o 'nada de acuerdo' con esta afirmación. Al amparo de este análisis, y si se considera el enunciado P4 ("En general mis conocimientos sobre MATLAB han mejorado"), se concluye que el conjunto completo de estudiantes está de acuerdo o muy de acuerdo en considerar que su conocimiento de MATLAB ha mejorado. Lo que indica que las actividades propuestas proporcionan una mejora considerable tanto en los usuarios poco iniciados en MATLAB como en los avanzados. Resulta natural que en las cuestiones P2 ("Tenía conocimientos previos de RoboDK") y P3 ("Tenía conocimientos previos de programación de Robots Manipuladores") la totalidad esté en desacuerdo, ya que no es habitual haber sido adiestrado antes en estas cuestiones, mientras que en la P5 ("En general mis conocimientos sobre RoboDK han mejorado") la totalidad de alumnado está de acuerdo con la afirmación.

b) Percepción sobre el uso del simulador. En esta área merece la pena comentar el enunciado P6 ("Los resultados en el simulador me han permitido entender las tareas a programar"), para ella la mediana es 5 y la media 4.5, lo que indica que la gran mayoría del estudiantado está de acuerdo con esta afirmación. Respecto a las cuestiones P8 ("Los resultados en el simulador me han permitido depurar errores en la programación"), P9 ("Los resultados en el simulador me han permitido planificar una tarea completa") y P10 ("Valoro positivamente poder revisar mis cálculos sobre un simulador antes de usar un robot real") los valores de las medianas (4, 4 y 5) indican que la mayoría (o en el caso de la P10 la totalidad) está de acuerdo con estas afirmaciones.

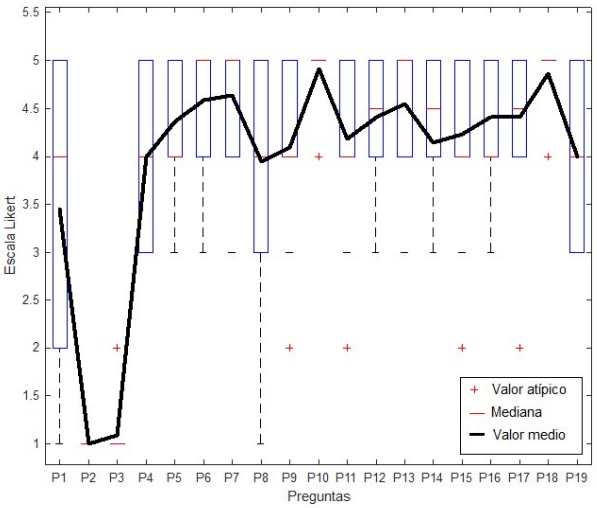


Figura 12: Resultados estadísticos de las encuestas

c) Percepción sobre el uso del robot real. Del análisis de las respuestas a los enunciados P11 ("El acceso al robot real es sencillo y seguro"), P12 ("Los resultados con el robot real me han permitido mejorar la comprensión sobre la tarea a realizar"), P13 ("Los resultados con el robot real me han permitido mejorar la percepción del espacio de trabajo") y P14 ("Los resultados con el robot real me han permitido mejorar el código programado"), y de los valores asociados a sus medianas (4, 4.5, 5 y 4.5) se colige que la gran mayoría del alumnado considera que el método propuesto para compartir el uso de los robots es sencillo, y que su utilización ayuda significativamente en la ejecución eficaz de las actividades de programación.

d) Percepción sobre el método seguido. El análisis de las respuestas a los enunciados P17 ("El orden de las actividades ha sido correcto, permitiéndome avanzar en la programación"), P18 ("Ver el resultado de mi programación en el simulador y su reflejo en la acción real del robot me ha resultado satisfactorio") y P19 ("Frente a otros métodos usados en otras asignaturas de la titulación, éste me ha resultado mejor"), y los valores de sus medianas (4.5, 5 y 4) permite concluir que la práctica totalidad del estudiantado está de acuerdo en que ha sido satisfactoria la experiencia de combinar el uso del simulador y del robot real tal y como se ha propuesto en este trabajo, que está satisfecho con el orden en que se han secuenciado las actividades, y que un 50% del alumnado considera que la metodología utilizada es mejor que la utilizada en otras asignaturas, no habiendo nadie que esté en desacuerdo con esa afirmación.

6. Conclusiones

En este trabajo se presenta una propuesta para la realización de actividades prácticas relativas al desarrollo de competencias para la programación de robots manipuladores. La estrategia docente se basa en el uso conjugado de simulación e implantación de la tarea programada en un sistema robótico real.

La base del sistema propuesto está soportada en el entorno proporcionado por el programa MATLAB, mediante el uso de un lenguaje diseñado particularmente para interaccionar con el programa RoboDK, con el fin de testar en simulación el código

generado, y posteriormente con el robot real, para comprobar físicamente que el robot efectúa con corrección las tareas programadas.

La propuesta se fundamenta en un conjunto de competencias básicas que engloban las capacidades y destrezas que desarrolla la inclusión de las actividades descritas. Estas competencias han sido elaboradas a partir del análisis de las competencias específicas y los resultados de aprendizaje de numerosas asignaturas de robótica impartidas en grados y másteres universitarios. En un futuro se plantea incorporar en el sistema el acceso a elementos periféricos y sensores, presentes en RoboDK o en el robot real, mediante la incorporación de nuevos comandos del lenguaje desarrollado.

Agradecimientos

Los autores agradecen al Departamento de Ingeniería Electrónica, de Sistemas Informáticos y Automática la facilidad proporcionada para la realización de este trabajo.

Referencias

Ajwad, S. A., Islam, R. U., & Iqbal, J., 2017. Exploring the Training Potential of Recent Virtual Robotic Platforms: A Comprehensive Review.Journal of Advancements in Robotics, 4(2),on-line.

Alhama Blanco, P. J., Abu-Dakka, F. J., & Abderrahim, M.,2018. Practical use of robot manipulators as intelligent manufacturing systems. Sensors, 18(9). DOI:10.3390/s18092877.

Álvarez, J.D., Muñoz, M., 2022. Guía docente de la asignatura Robótica. Grado en Ingeniería Electrónica Industrial. Universidad de Almería. https://www.ual.es/estudios/grados/presentacion/plandeestudios/asignatura/4310/43104216.

Barrientos, A. Peñin, L.F., Balaguer, C, Aracil, R., 2007. Fundamentos de robótica. Mcgraw hill.

Bauzano, E., Muñoz, V. F., & Garcia-Morales, I., 2010. Auto-guided movements on minimally invasive surgery for surgeon assistance. In 2010 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 1843-1848. DOI: 10.1109/IROS.2010.5650875

C. -C. Hu, H. -T. Tseng, M. -H. Chen, A. Goh Phaik Imm and N. -S. Chen, 2020. Comparing the effects of robots and IoT objects on STEM learning outcomes and computational thinking skills between programming-experienced learners and programming-novice learners, 2020 IEEE 20th Int. Conf. on Advanced Learning Technologies (ICALT), 2020, pp. 87-89. DOI: 10.1109/ICALT49669.2020.00033.

Cabanes, I., Mancisidor, A., 2022. Guía docente de la asignatura Robótica Industrial. Master en Ingeniería de Control, Automatzación y Robótica. U. País Vasco. https://www.ehu.eus/es/web/master/master-ingenieria-control-automatizacion-robotica/materia?p_anyo_ofd=20220&p_anyo_pop=20140&p_cod_centro=345&p_cod_materia=8093&p_cod_asignatura=504137&p_tipo_asignatura=1.

Craig, J. J. (2022). Introduction to robotics: Mechanics and Control (4th Global Edition). Pearson Educacion.

Chakraborty, S., & Aithal, P. S., 2021. Forward and Inverse Kinematics Demonstration using RoboDK and C. International Journal of Applied Engineering and Management Letters (IJAEML), 5(1), 97-105. DOI: 5281/zenodo.4939986

Corke, P., 2017. Robot manipulator capability in MATLAB: A tutorial on using the robotics system toolbox. IEEE Robotics & Automation Magazine, 24(3), 165-166. DOI: 10.1109/MRA.2017.2718418

Corke, P., 2017. Robotics, vision and control: fundamental algorithms. In: MATLAB Second, Completely Revised, vol. 118. Springer, Cham.

Galindo Andrades, C., Gonzálo Monroy, J., 2022. Guía docente de la asignatura Control y Programación de Robots, Grado en Ingeniería Electrónica, Robótica y Mecatrónica, Universidad de Málaga https://oas.sci.uma.es:8443/ht/2022/ProgramasAsignaturas_Titulacion_5156_AsigUMA_52918.pdf.

Gambao Galán, E., 2022. Guía docente de la asignatura Robótica Aplicada, Master Universitario en Robótica y Automática. Universidad Politécnica de Madrid. https://www.upm.es/comun_gauss/publico/guias/2022-23/1S/GA_05BH_53001559_1S_2022-23.pdf.

Gámez García, J., Cano, P., 2022. Guía docente de la asignatura Robótica Industrial. Grado en Ingeniería Electrónica Industrial. Universidad de Jaén. https://uvirtual.ujaen.es/pub/es/informacionacademica/catalogoguiasdocentes/p/2021-22/4/131A/13113009/es/2021-22-13113009_es.html.

García-Morales, I., Serón, J. 2022. Guía docente de la asignatura Fundamentos de Robótica, Grado en Ingeniería Electrónica, Robótica y Mecatrónica, U. Málaga.https://oas.sci.uma.es:8443/ht/2022/ProgramasAsignaturas_Titulacion_5156_AsigUMA_52913.pdf.

Gil, A., Reinoso, O., Marin, J. M., Paya, L., & Ruiz, J., 2015. Development and deployment of a new robotics toolbox for education. Computer Applications in Engineering Education, 23(3), 443-454. DOI: 10.1002/cae.21615.

Gil, A., Sabater, J.M., 2022. Guía docente de la asignatura Robótica. Grado en Ingeniería Electrónica y Automática Industrial. Universidad Miguel Hernández. https://umh1770.umh.es/.

González Victores, J.C., 2022. Guía docente de la asignatura Robótica Industrial. Grado en Ingeniería Electrónica Industrial y Automática. U. CarlosIII. https://www.uc3m.es/grado/electronica#programa

Heiberger, R., & Robbins, N., 2014. Design of Diverging Stacked Bar Charts for Likert Scales and Other Applications. Journal of Statistical Software, 57(5), 1–32. DOI: 10.18637/jss.v057.i05.

Holubek, R., Delgado Sobrino, D. R., Košťál, P., & Ružarovský, R., 2014. Offline programming of an ABB robot using imported CAD models in the RobotStudio software environment. Applied Mechanics and Materials, 693,62-67. DOI:10.4028/www.scientific.net/AMM.693.62

Huang, L., Varnado, T., & Gillan, D., 2013. Practices of teaching problem solving skills in robotics education. In Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 57 (1),1830-1834. DOI:10.1177/1541931213571409.

Lora, J.S., Castaño, J.A., 2022, Guía docente de la asignatura Robótica Industrial. Grado en Ingeniería de Robótica Software, Universidad Rey Juan Carlos. https://gestion3.urjc.es/guiasdocentes/

Lu, L., 2019, "ME 625-101: Introduction to Robotics". Mechanical and Industrial Engineering Syllabi, New Jersey Institute of Technology. https://digitalcommons.njit.edu/mie-syllabi/129

Lukač, D., 2018. Simulation of a pick-and-place cube robot by means of the simulation software KUKA Sim Pro. In 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 0846-0849.

Martín Rico, J.C., 2022, Guía docente de la asignatura Arquitectura de Software para Robots. Grado en Ingeniería de Robótica Software, U. Rey Juan Carlos. https://gestion3.urjc.es/guiasdocentes/

Maza, J. I., & Ollero, A., 2001. Hemero: a MATLAB-simulink toolbox for robotics. In 1st Workshop on Robotics Education and Training, 43-50. DOI: 10.23919/MIPRO.2018.8400156

Mondada, F., Bonani, M., Raemy, X., Pugh, J., Cianci, C., Klaptocz, A., Martinoli, A.,2009. The e-puck, a robot designed for education in engineering. In Proceedings of the 9th conference on autonomous robot systems and competitions, 1, 59-65.

Muños de la Peña, D. Domínguez, M., Gomez-Estern, F., Reinoso, Ó., Torres, F., Dormido, S., 2022. Estado del arte de la educación en automática. Revista Iberoamericana de Automática e Informática Industrial, 19(2), 117-131. DOI:10.4995/riai.2022.16989

Newton, K. J., Leonard, J., Buss, A., Wright, C. G., & Barnes-Johnson, J., 2020. Informal STEM: Learning with robotics and game design in an urban context. Journal of Research on Technology in Education, 52(2), 129-147. DOI:10.1080/15391523.2020.1713263

Ollero Baturone, A. (2005). Robótica: manipuladores y robots móviles. Marcombo.

Torres, F.,Pomares J., Gil P, Puente S.T., Aracil, R., 2002. Robots y sistemas sensoriales, Prentice Hall.

Romero Tello, A., Gracias Heras,C., 2022, Guía docente de la asignatura Robótica Industrial. Grado en Ingeniería Electrónica y Automática. U. de Zaragoza.https://sia.unizar.es/documentos/doa/guiadocente/2022/29827_es.pdf.

Valera, A., Soriano, A., & Vallés, M., 2014. Plataformas de bajo coste para la realización de trabajos prácticos de mecatrónica y robótica. Revista Iberoamericana de Automática e Informática Industrial, 11(4), 363-376. DOI:10.1016/j.riai.2014.09.002.

Vivas, A., & Sabater, J. M., 2021. Ur5 robot manipulation using matlab/simulink and ROS. Int. Conf. on Mechatronics and Automation. DOI:10.1109/ICMA52036.2021.95126